

#10. Security and cryptography



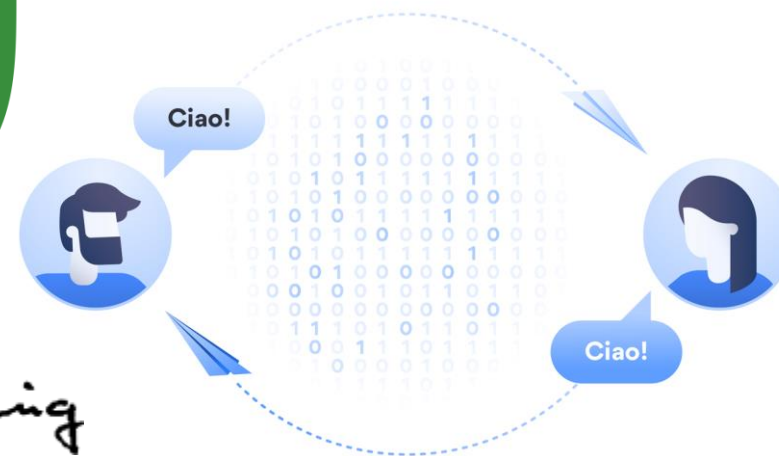
University of Bergamo
Master Course in Project and
Management of Tourism Systems
Academic Year 2021-2022
IT for Tourism Services



What are we talking about this time?



1. Privacy vs. security
2. Secure connections
3. Secure websites
4. Malware and hacking
5. Cryptography
6. The GDPR, again
7. Alan Turing



A. M. Turing

image credits to דרׁון פּורטוגלי, Pixlr, Comodo Enterprise and NordVPN

It's not mainly about privacy, here



One of our meetings must specifically deal with **security**, as it is globally defined. (Italian speakers among us might call the problem **sicurezza**.)

Security is often mistaken as **privacy**.

It is true that, generally speaking, a secure network connection contributes to the privacy of users who share some content.

The two issues, however, are different.

Privacy is a **legal** issue.

Security is a technical issue.

The technical condition is **encryption**.



image credits to [Forbes India](#) and [ehorus.com](#)

Encrypted messages? Not only.



The concept behind **encryption** is quite simple – make the data illegible for everyone else except those specified.

This is done using **cryptography** – the study of sending messages in a **secret form** so that only those authorized to receive the message are able to read it.

The easy part of encryption is applying a **mathematical function** to something and making it encrypted.

The harder part is to ensure that the people who are supposed to decipher the message can do so with ease, yet **only those authorised** are able to decipher it.

Let's be careful, however.

Internet **security** deals with encrypted **connections**, not encrypted messages.

Dangerous websites

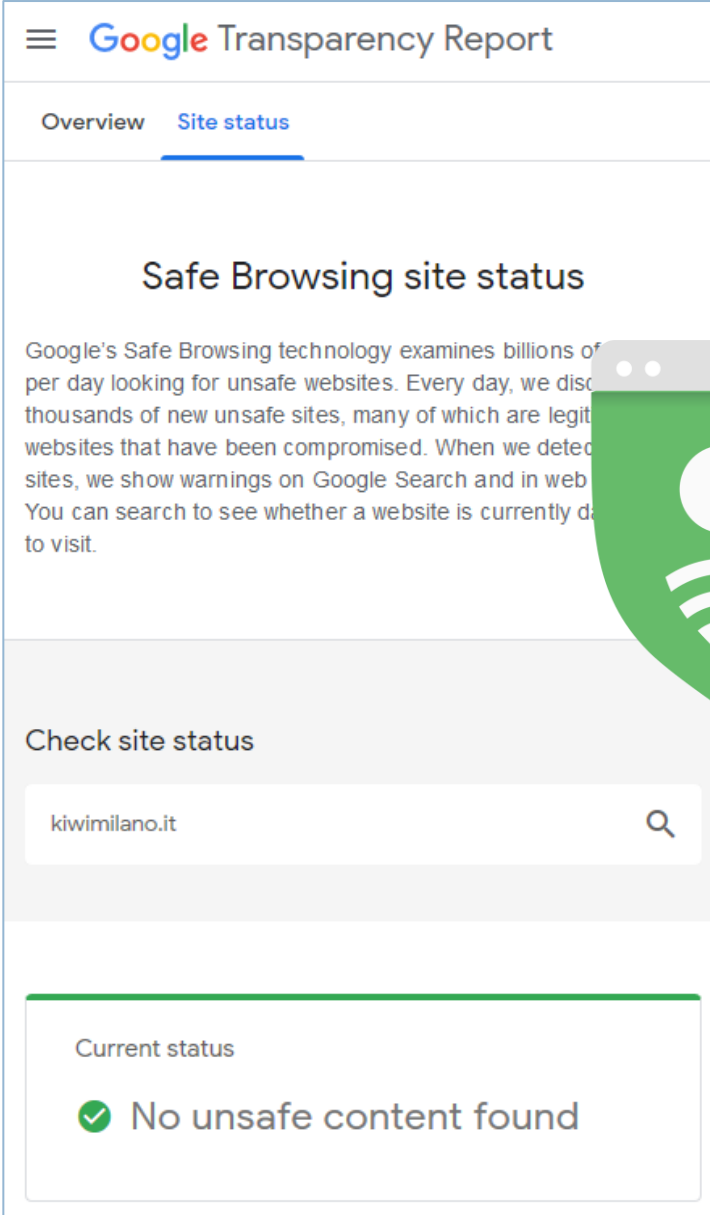
A possible confusion.

It may seem that, as for other functions, the Google system provides a solution thanks to one of their **online tools**: a free network tool.

The online tool that may seem able to verify security is the **Google Safe Browsing site status**.

It is not so. The Google Safe Browsing site status analyses the **security** of the **content** of a website.

Content security is **different** from security.



The screenshot shows the Google Transparency Report interface. At the top, there is a navigation bar with the Google logo and the text 'Transparency Report'. Below this, there are two tabs: 'Overview' and 'Site status', with 'Site status' being the active tab. The main heading is 'Safe Browsing site status'. The text below explains that Google's Safe Browsing technology examines billions of websites per day for unsafe content. A search bar labeled 'Check site status' contains the URL 'kiwimilano.it'. Below the search bar, a green box indicates the 'Current status' as 'No unsafe content found' with a green checkmark icon.



Malware



The attack implemented by Mafiaboy – a story we mentioned when reading **Linked** – was based on an **infection** that Mafiaboy spread in tons of computers.

That malware was designed to start a request for denial of service from tons of computer in a single moment. This resulted in a instant overload and a consequent **crash** (a collapse) of the Yahoo! server – and several other servers a bit later.

Those servers hadn't been “infected” by their managers! It was Mafiaboy who did so.

Well... **Computers** may become “infected” when their browsers visit websites **containing files with unsought, concealed, and malicious instructions.**



Hacking



Mafiaboy was in fact a hacker, who later would find a job as a specialist in computer security. Today, most hackers act by making sure that users' computers, visiting an "infected" site, are surreptitiously put in contact with another "infective" server. In such cases, the user remains completely unaware of what happens.

What the [Google Safe Browsing site status](#) does is verifying that on the server of a website – the one which the Google Safe Browsing site status is asked to verify – there are no files with unsought, concealed, and malicious instructions.

In short, that there is [no malware](#).

It is essentially a verification similar to those that antivirus software make at every request.



Encrypted connection



Returning to security, here's the crucial point to consider.



Ciao!

Security encrypts the connection between one computer and another. The message cannot be deciphered because it is sent through a “waterproof” connection.



Ciao!

image credit to NordVPN

Padlocks & https



Encrypted connections use a different transmission protocol. Webpages – or e-mail messages – are connected through **https** instead of **http**.

An **encrypted layer** is placed between the server and the client browser.



Transactions' security



The **https** protocol has been used by **financial websites** for a long time.

The **e-commerce** process used to begin through a non encrypted connection (visiting an hotel website, for instance), then **moving to a transaction processor** connected through an **encrypted** connection to a **financial website** (the website of a credit card company, for instance), when paying.

Once the payment was accepted, the transaction processor sent **the user back** to the originating non encrypted connection, to **inform the original website** that the transaction had been completed.

What granted that the user was the same between the financial and the originating websites?

Cookies, of course!

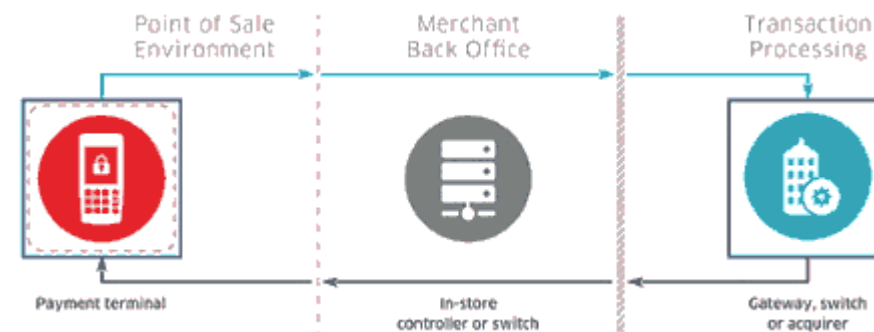
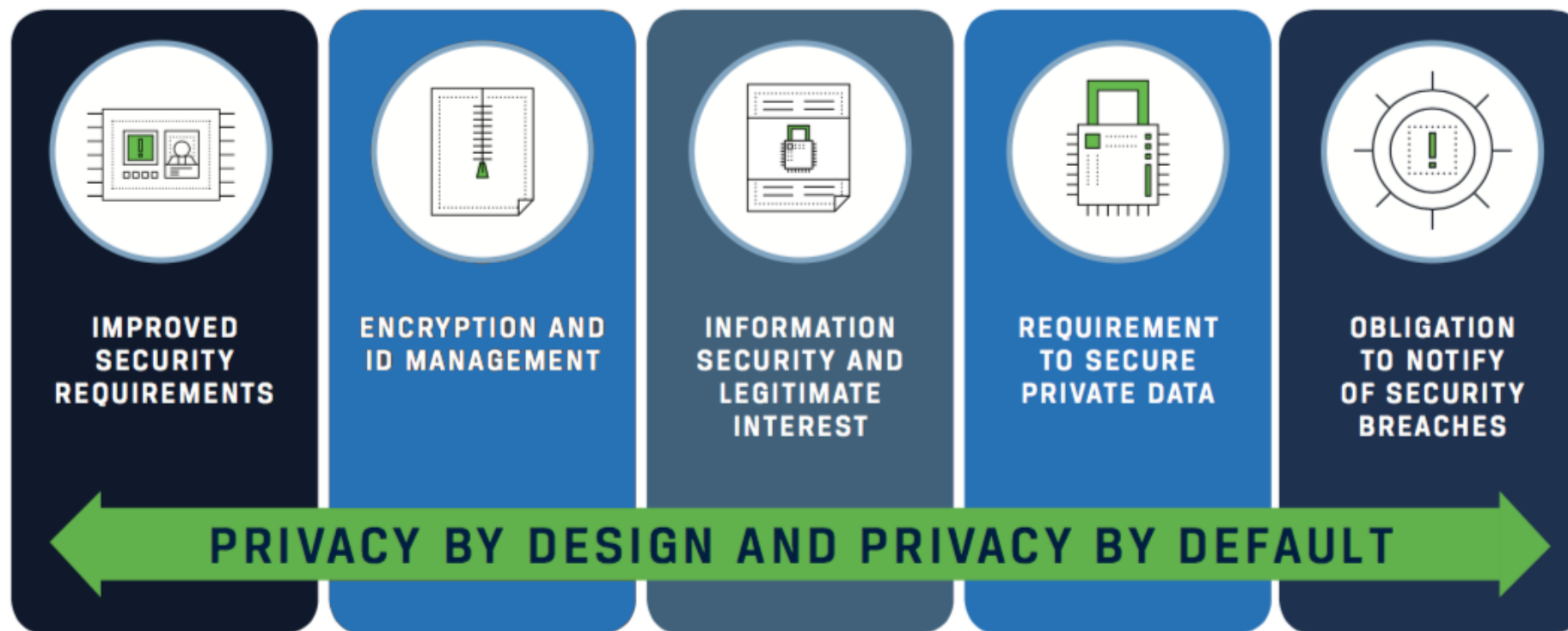


image credit to **North American Bancard**

The GDPR, again



The process is the same today, but the **GDPR** (EU General Data Protection Regulation) requires that the **originating website**, too, is **connected** through **https**.



Google itself penalizes websites which do not connect through **https**. Today **https** is the **default**.

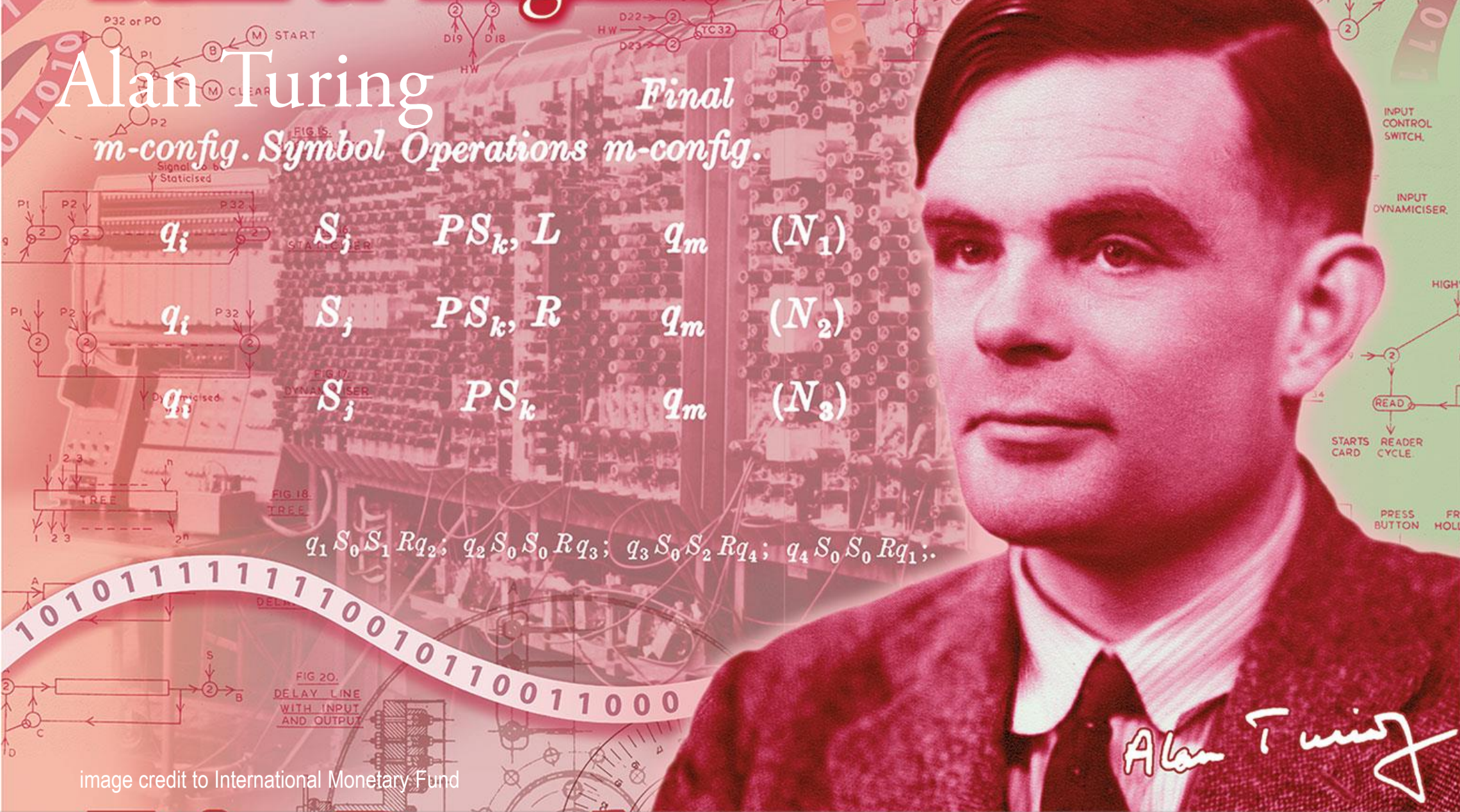
image credit to bankinfosecurity.com

Bank of England

Alan Turing

Final m-config. Symbol Operations m-config.

Final



q_i S_j PS_k, L q_m (N_1)



q_i S_j PS_k, R q_m (N_2)



q_i S_j PS_k q_m (N_3)



$q_1 S_0 S_1 R q_2; q_2 S_0 S_0 R q_3; q_3 S_0 S_2 R q_4; q_4 S_0 S_0 R q_1;$



Alan Turing

Alan Turing (1912-1954)

image credit to International Monetary Fund



The father of computer science



Introducing encryption, we cannot fail to emphasize that a story related to encryption was fundamental in the history of computer science.

Here's a relevant text from Wikipedia.

“Alan Mathison Turing (23 June 1912 – 7 June 1954) was an English mathematician, computer scientist, logician, cryptanalyst, philosopher, and theoretical biologist.

Turing was highly influential in the development of theoretical computer science, providing a formalisation of the concepts of algorithm and computation with the Turing machine, which can be considered a model of a general-purpose computer. Turing is widely considered to be the father of theoretical computer science and artificial intelligence.

During the Second World War, Turing worked for the Government Code and Cypher School (GC&CS) at Bletchley Park, Britain's codebreaking centre that produced Ultra intelligence.”

Bletchley Park



“Here, he devised a number of techniques for speeding the breaking of German ciphers, including improvements to the pre-war Polish bombe method, an electromechanical machine that could find settings for the Enigma machine. Turing played a crucial role in cracking intercepted coded messages that enabled the Allies to defeat the Axis powers in many crucial engagements, including the Battle of the Atlantic.

Turing was prosecuted in 1952 for homosexual acts. He accepted chemical castration treatment, with DES, as an alternative to prison. Turing died in 1954, 16 days before his 42nd birthday, from cyanide poisoning. An inquest determined his death as a suicide.”

It is an irony that a mathematician who was so important for the defeat of deadly ideologies was almost literally killed by the laws of a parliamentary democracy.

A. M. Turing

Key points

1. Security is not privacy
2. ... and not encrypted messages
3. What Google reveals
4. Secure is the connection
5. What the padlock reveals
6. Secure Europe
7. A mathematician

Final

m-config. Symbol Operations m-config.

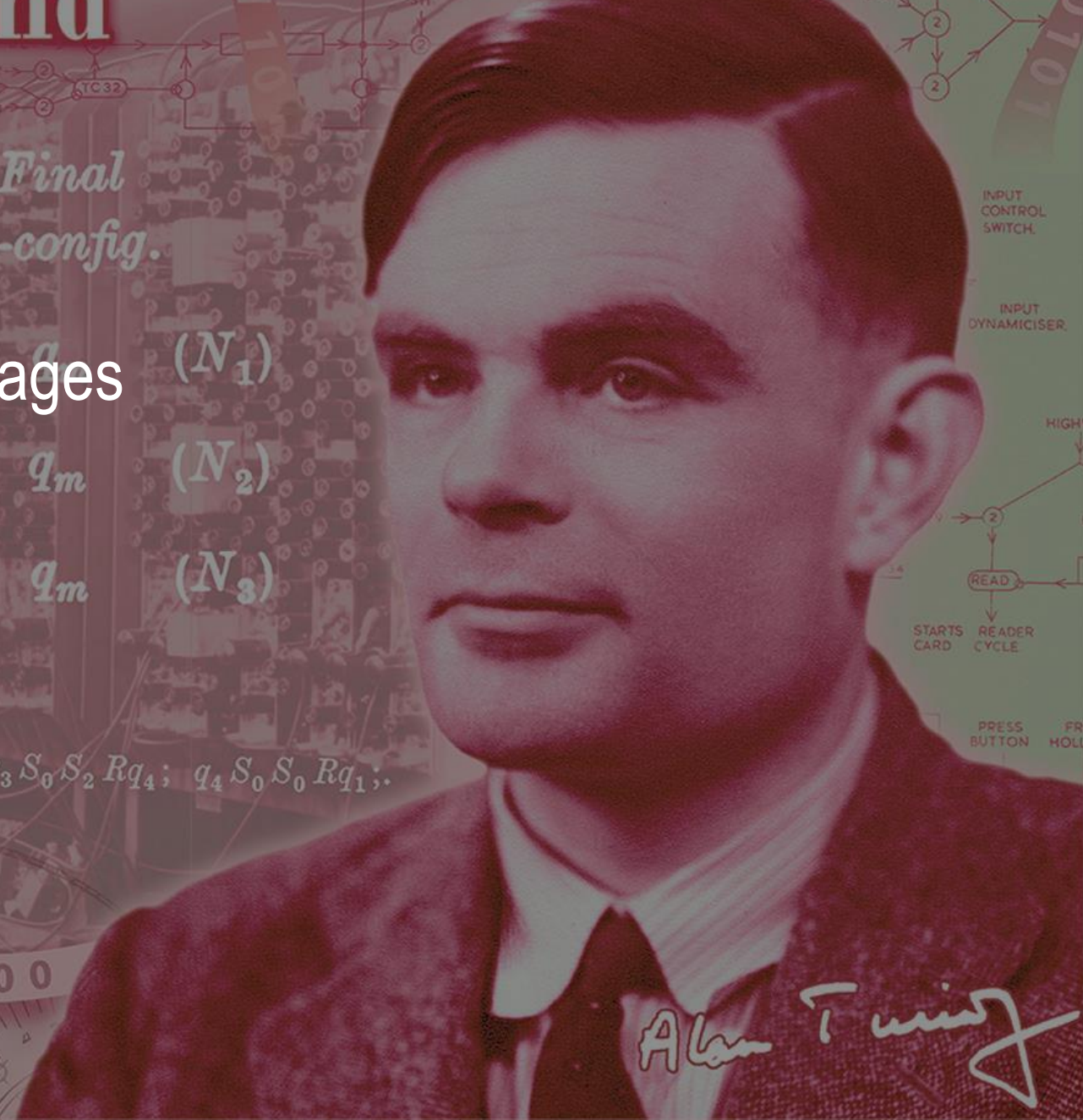
(N_1)

(N_2)

(N_3)

$q_1 S_0 S_0 R q_1; q_2 S_0 S_0 R q_3; q_3 S_0 S_2 R q_4; q_4 S_0 S_0 R q_1;$

image credit to International Monetary Fund



Alan Turing (1912-1954)